National Soil Information System 5.2
Database Structure Guide

United States Department Of Agriculture
Natural Resources Conservation Service
Information Technology Center
2150 Centre Ave., Bldg. A
Ft. Collins, CO 80526-1891
(970) 295-5464

NASIS Project
Project Manager: Ken Harward
Soil Survey Business Area Project Manager: Terry Aho
Technical Project Manager: Gary Spivak

Introduction

The Purpose of this Document

The purpose of this document is to describe the conventions used in the NASIS database and aid in the understanding of its structure. A better understanding of the underlying NASIS database structure will hopefully make it easier to understand and work with NASIS data.

The NASIS database is a relational database implemented in Informix On-Line. There have been 9 versions of NASIS, 1.0, 2.0, 3.0, 3.1, 4.0, 4.1, 5.0, 5.1 and now 5.2.

This document should be used in conjunction with a set a reports that detail various aspects of the database (see NASIS Template Model Reports). This document is not intended to explain the workings of the NASIS software itself.

Viewing the NASIS database with the NASIS Software

The tables in a NASIS database are managed in groups. We refer to one of these groups as an "owned object" (see Owned Objects). A table in NASIS can belong to one and only one owned object.

One function of the NASIS software is to retrieve data from a NASIS database for viewing and editing. When viewing data in the NASIS software, data for only one owned object is visible at a time.

When you view data in NASIS, that data is separated into individual spreadsheet windows where each spreadsheet window corresponds to a particular table in the underlying NASIS database. Multiple spreadsheet windows are always cascaded. When more than one window is cascaded, only one record is visible in any window other than the bottom window. The visible records in the windows other than the bottom window constitute the parent record hierarchy of all of the records, if any, visible in the bottom window.

The columns displayed in a table's spreadsheet window may not be exactly the same columns that exist in that table in the underlying NASIS database. There are a number of reasons why this is the case.

A table's spreadsheet window may or may not display all of the columns in the underlying database table. Some of the underlying columns are purposely hidden at all times. Other columns may be hidden because the current NASIS session may be using an Edit Setup that excludes one or more columns from being displayed.

An Edit Setup is a set of specifications for the following:

> Which columns in a table should be displayed in that table's corresponding spreadsheet window?
>
> In what order should the selected columns of a table be displayed?
>
> What is the display width of each of the selected columns?

How many of the left most selected columns of a table should be frozen (not subject to scrolling)?

On what column or columns should the records in a table be sorted?

Prior to NASIS 4.0, a NASIS user had no control over any of these choices. As of NASIS 4.0, a NASIS user could customize which of a table's columns are visible in its corresponding spreadsheet window.

There are two other reasons that the columns displayed in a table's spreadsheet window may not be exactly the same columns that exist in that table in the underlying database. One reason has to do with how "choice" attributes are handled in NASIS. The other reason has to do with how primary and foreign keys are handled in NASIS.

A "choice" attribute in NASIS is an attribute whose values are restricted to a finite set of choices or codes. The set of values that a choice attribute can assume is referred to as that attribute's domain. NASIS has a somewhat complex scheme for handling domains. At this point, suffice it to say that what is displayed by the NASIS software as the value for a choice attribute is not what is actually stored in the underlying database table (see Class Attributes).

A similar situation exists for foreign keys. A foreign key is a column or set of columns in a table that uniquely identify one and only one record in a related table. In NASIS, all foreign keys (and all primary keys too for that matter) are meaningless integer numbers that are assigned by the NASIS software (see Record Identification). In a table's spreadsheet window, the integer numbers that make up a foreign key are not displayed. Instead, some meaningful business related columns from the related table are displayed in place of the foreign key (see Links).

Summary of NASIS 5.2 Database Changes

For NASIS 5.2, the only actual database changes were:

A sodium fluoride pH column was added to the pedon horizon table.

A pedon horizon feature color table was added.

An ICAMS ID column was added to the NASIS user table, to facilitate a NASIS user's access to the Soil Data Warehouse Staging Server without having to login to NASIS after having logged in via WebCAAF.

A new column was added to the "dd_tbl" data dictionary table to indicate a table's level in the directed acyclic graph that represents the NASIS database. This change is transparent to NASIS end users.

The main reason that NASIS 5.2 was released was to provide the capability to be able to export data to the Soil Data Warehouse. There were a variety of application related enhancements that are not related to the underlying database structure.

NASIS Metadata is Available from the NASIS Web Site

This document and its associated reports are available for viewing or printing from the NASIS web site. The URL of the general NASIS web site is:

http://nasis.nrcs.usda.gov

The URL from which this document and related documents may be viewed and printed is:

http://nasis.nrcs.usda.gov/documents/metadata/5_2

The on-line versions of these documents are stored in “pdf” format. This means that a user must have Adobe Acrobat Reader installed in order to view and print these documents. The Adobe Acrobat Reader is a free web browser plug-in that may be downloaded from the following URL:

http://www.adobe.com/products/acrobat/readstep2.html

The Actual Physical NASIS Database

Tables in a NASIS database fall into one of four categories, soil business related tables, data dictionary tables, edit tables or soil metadata repository tables. A soil business related table is any table used to permanently record National Cooperative Soil Survey (NCSS) related information. Data dictionary tables are those tables used to control the workings of the NASIS software and the NASIS security system. Throughout the NASIS documentation, the terms “data dictionary”, “operational data dictionary” and “ODD” are used interchangeably. Soil metadata repository tables record descriptions of the entities and attributes that make up various soil survey related information systems, including NASIS and SSURGO. Any soil business related table, data dictionary table or soil metadata repository table whose contents can be altered via the NASIS software requires a corresponding edit table. Edit tables are where the changes a NASIS user makes to a table are stored until such time as the NASIS user either chooses to commit those changes to the permanent tables or discard the changes entirely.

Soil Business Related Tables

A graphical view of the relationships between the various soil business related tables is available in the NASIS on-line help system and as separate documents that may be viewed or printed from the NASIS web site:

http://nasis.nrcs.usda.gov/documents/metadata/5_2

The soil business related tables are split into two diagrams:

Traditional Aggregate Data Structure Diagram (Legend/Data Mapunit)

Traditional Point Data Structure Diagram (Site/Pedon/Transect/Site Association)

These diagrams do not show the columns that make up each soil business related table, only the relationships between those tables. The description and contents of a particular table can be found in the Table Structure Report (see NASIS Template Model Reports).

Edit Tables

Prior to NASIS 4.0, the edit tables were a permanent part of an actual physical NASIS database, and each concurrent NASIS user shared the same set of edit tables. Since NASIS 4.0, each NASIS user has their own private set of edit tables, which are created when they invoke NASIS, and are dropped when they terminate their session.

The physical name of a table's corresponding edit table is the same as the physical name of the table, but with an "e_" concatenated to the front of the name. The name of the Component table's corresponding edit table is "e_component", since the physical name of the Component table is "component".

Any change a NASIS user makes to a table during an edit session is temporarily saved to that table's corresponding edit table. If a NASIS user chooses to terminate their edit session without saving any changes, any records in the edit tables generated by that NASIS user during their session are discarded. Each edit table contains columns in addition to those of its corresponding table. These columns are used to relate a record in an edit table to its corresponding record in the permanent NASIS database, and to keep track of the status of that record (Has it been modified? Has it been marked for deletion?).

The edit tables are only meaningful to an active NASIS edit session. There is little reason that anything other than the NASIS software would be interested in the contents of these tables. A general description of these tables is included merely to completely document all of the tables present in an actual physical NASIS database.

Operational Data Dictionary Tables

The NASIS software is table driven. What is displayed on screen in NASIS is controlled by the contents of the ODD tables. The data dictionary tables in NASIS could describe an auto-parts database as easily as they describe a soil database. To a large degree, the NASIS software has no idea of what type of database that NASIS is. Of course this generalization is only true to a point. Some functions in NASIS, like the export functions, look for particular tables to exist in the database. Think of the data dictionary tables as the minimal set of tables required in order for the NASIS software to work.

Some of the data dictionary tables are static and cannot be modified short of releasing a new version of NASIS. These static data dictionary tables cannot be viewed or edited in NASIS. The static data dictionary tables may however be queried and reported on. These are the tables that define the structure of the underlying NASIS database.

A number of other data dictionary tables may be viewed and edited in NASIS. One set of data dictionary tables that may be viewed and edited in NASIS are those that pertain to the NASIS security system. These tables include the NASIS Site, Group, Group Member and NASIS User tables. The contents of these tables determine who can access a NASIS database and what data they can create or modify, if any.

Other data dictionary tables that may be viewed and edited in NASIS are those that pertain to the creation and maintenance of a variety of NASIS system owned objects. These owned objects include Queries, Reports, Properties, Evaluations, Rules, Calculations, Edit Setups and Distribution Metadata.

Soil Metadata Repository Tables

Prior to NASIS 5.1, the soil metadata repository was a NASIS database separate from the NASIS soil database. The soil metadata repository tables were merged into NASIS 5.1 because we got tired of having to update two different databases every time there was a new version of NASIS. The addition of the soil metadata repository tables to NASIS is transparent to general NASIS users because the tables that make up the Soil Metadata Repository are only visible to members of NASIS site "Soil Metadata Repository".

Owned Objects

For the purpose of locking and access control, tables in NASIS are managed in groups. Each of these groups of tables is referred to as an "owned object". A table in NASIS may be a member of one and only one owned object. Each editable table in NASIS must be a member of an owned object. A non-editable table may or may not be a member of an owned object. When a NASIS user loads an instance of an owned object into NASIS, all records that make up that instance of that particular owned object are locked. NASIS contains the following 25 owned objects:

Owned Object Name	Description
Area Type	The list of geographic area types and areas that may be referenced in NASIS. The master list is provided by the NSSC, but NASIS sites may add their own area types and areas.
Legend	Corresponds to a soil survey area. NASIS can record more than one vintage of the same soil survey area. This owned object includes the map unit soil entity.
Data Mapunit	The data used to characterize a map unit. This owned object includes the component and horizon soil entities. In NASIS, map units and the data that characterize them are separated, in order to facilitate perfect joins across arbitrary political boundaries.
Site Association	A group of related Sites.
Site	A point location where a pedon was described or other ecological site related data was collected.
Pedon	The morphological description of a soil at a point on the ground.
Transect	A group of related pedons taken to determine the variability of the soil across a landscape.
Local Plant	An instance of a plant whose common name may be different than the common name recorded in the official Plants database. Referenced by various map unit component and site tables that record vegetation. This owned object serves as a lookup table in NASIS.
Plant	An instance of a plant from the official Plants database. This owned object serves as a lookup table in NASIS. Only members of the NASIS site "flora" can create, modify and delete instances of this owned object.
Ecological Site	An instance of an official NRCS ecological site from the official Ecological Site Description database. This owned object serves as a lookup table in NASIS. Only members of the NASIS site "flora" can create, modify and delete instances of this owned object.
Other Vegetative Classification Type	A non-NRCS ecological site or plant community classification. This owned object serves as a lookup table in NASIS.
Geomorphic Feature Type	The list of geomorphic feature types and geomorphic features that may be referenced in NASIS. This owned object serves as a lookup table in NASIS. Only members of the NASIS site "pangaea" can create, modify and delete instances of this owned object.
Distribution Metadata	The set of metadata that includes the selection criteria and list of legends, map units, components, interpretations and kinds of narrative text included in a new SSURGO export.
Query	A script for retrieving data from a NASIS database for viewing, editing and reporting during a NASIS session.
Report	A script for retrieving and formatting of data from a NASIS database for output to a hard copy Report or file.
Property	A virtual attribute that is calculated on demand during report processing or interpretation generation.
Evaluation	A function that transforms the value of a Property into a fuzzy logic value in the range zero to one.

Rule	A function of one or more Evaluations and/or other Rules that produces a fuzzy logic value in the range zero to one. A Rule corresponds to an "interpretation" in NASIS.
Calculation	A script for calculating a value for one or more columns, based on the value of one or more other columns. Alternatively, a script that determines if the value of one or more columns appears to be valid. Only members of the NASIS site "pangaea" can create, modify and delete instances of this owned object.
Edit Setup	A set of specifications that describe what data should be visible during a NASIS session, and how that data should be ordered and sorted.
NASIS Site	An organization unit responsible for the creation and maintenance of a subset of the data in NASIS.
NASIS User	Identifies a valid NASIS user. This owned object was split out from the NASIS Site owned object in NASIS 5.0.
Information System	A soil survey related information system, such as NASIS, SSURGO, Windows Pedon, etc. This owned object is only visible to members of NASIS site "Soil Metadata Repository".
Domain Group	A set of domains referenced by one or more Information Systems. This owned object is only visible to members of NASIS site "Soil Metadata Repository".
Unit of Measure	A unit of measure in which an attribute value is recorded. This owned object is only visible to members of NASIS site "Soil Metadata Repository".

The NASIS Template Model

NASIS Template Model Overview

The reports that accompany this document describe what we refer to as the NASIS Template Model.

Warning: There is not necessarily a one to one correspondence between the columns described in the NASIS Template Model and the columns in an actual physical NASIS database.

The reason for this is that we use a shorthand notation for two specific types of columns, "modal" and "calculable" columns. These types of columns, which are listed as a single column in the NASIS Template Reports, actually expand into multiple columns in a physical NASIS database. Nevertheless, it is possible to unambiguously extrapolate the exact contents of a physical NASIS database by understanding how these two types of special columns are handled.

In order to characterize the range in characteristics of a soil map unit, many attributes are recorded as a range with an accompanying representative value. Such an attribute is what we refer to as a "modal" attribute. In the actual physical NASIS database, a modal attribute requires three columns, one for the low value of the range, one for the high value of the range and one for the representative value of the range. In the NASIS Template Model, a modal attribute is recorded as a single column, but associated with each attribute in the NASIS Template Model is a field that indicates whether or not that attribute is modal (see Modal Attributes).

Since version 3.0 of NASIS, an attribute may have an associated routine that will calculate its value based on the value of one or more other attributes. In the actual physical NASIS database,

a calculable attribute requires two columns, one for the value of that attribute and another accompanying status column that indicates if the current value was calculated, manually entered since being able to be calculated, manually entered prior to being able to be calculated, or if its origin is unknown. In the NASIS Template Model, a calculable attribute is recorded as a single column, but associated with each column in the NASIS Template Model is a field that indicates whether or not that column may be calculated (see Calculable Attributes).

NASIS Template Model Reports

The NASIS Template Model Reports cover all soil business related tables and all editable data dictionary tables. The non-editable data dictionary tables, edit tables and the soil metadata repository tables are not documented in these reports because they are of no interest to a general NASIS user.

The soil business related and editable data dictionary tables and attributes are not grouped by kind. All tables and attributes, regardless of type, are reported in alphabetical order, and are therefore intermingled.

Attribute Report

The Attribute Report lists attributes, independent of the tables in which they occur. An attribute may appear as a column in one or more tables. The Attribute Report includes attribute names, logical data type, minimum, maximum, attribute definition, and, when appropriate, the domain (choice list) name associated with a class (choice or coded) attribute (see Class Attributes). This is the only report where an attribute is defined or where its associated domain, if any, is listed.

Choice List Report

The Choice List Report lists the currently valid and obsolete choices in each domain (choice list). More than one attribute may share the same domain. A particular attribute may be associated with one and only one domain. An obsolete choice is not displayed in a choice list, but an obsolete choice may be entered in NASIS by typing in its code (Choice) directly, even though that code is not visible in the choice list. In other words, to enter an obsolete code, one must be aware that such a code exists, since it is not visible in the choice list. The only document that lists obsolete codes is the Choice List Report.

For class attributes of soil business related tables, the integer number associated with a choice (Choice ID), rather than the choice itself (Choice), is stored in the actual physical NASIS database. The Choice List Report also shows the Record ID associated with the choice list as a whole (Choice List ID). Using Choice List ID's greatly simplifies raw SQL queries (non-NASIS Queries) against class attributes (see Class Attributes).

Table Structure Report

The Table Structure Report lists the columns in a table, and some of the attributes of those columns. This report also shows the name of the ASCII delimited file associated with each NASIS table. These ASCII delimited files are used for export and import operations to information systems external to NASIS, the SSURGO template database for example.

The column titled “Sequence” indicates the default order in which the columns of a table will be displayed in that table’s corresponding spreadsheet window, not the physical order of the columns in the table. In a relational database, the physical order of the columns in a table is theoretically immaterial. Physical order is only a concern when unloading data from a database, and when loading data into a database.

Modal attributes (high, low, representative value attributes) and calculable attributes are listed as a single column (i.e. single row in the report), even though they are ultimately implemented as multiple physical columns. The contents of the report column titled “Exp. Var” (expression of variability) indicates whether a column is a modal, i.e. is implemented as a high, low and representative value. The contents of the report column titled “Calc?” indicates whether a column’s value may be automatically calculated.

Index Report

The Index Report lists indexes and the columns that make up each index. The Index Report is the only document that shows what combination, or combinations, of columns in a table must be unique (see Record Identification).

Relationship Report

The Relationship Report shows on what columns two related tables are joined. It indicates what table corresponds to the parent (the left side of the one to many or one to one relationship) and what table corresponds to the child (the right side of the one to many or one to one relationship). The vast majority of relationships in NASIS are one to many. There are only a couple of one to one relationships in the NASIS database.

Link Report

The Link Report shows the information that is used to resolve a lookup between two tables in the NASIS software. This report is most useful in understanding what takes place in a table’s spreadsheet window when that window displays information from a related database table (see Links).

Traditional Data Structure Diagrams

The Traditional Data Structure Diagrams are the traditional colorful diagrams that we have distributed with every release of NASIS. A Data Structure Diagram shows the relationships between soil business related tables, where all tables that belong to the same owned object are the same color. There are two NASIS Data Structure Diagrams, one for the aggregate or map unit data and one for the point or site/pedon data.

Technical Data Model Diagrams

A data model is a technical diagram that portrays the relationship either between logical entities or between tables in a database. A data model is more rigorous than a data structure diagram. The term “data model” is widely recognized and understood. The

term “Data Structure Diagram” has no widely recognized common meaning. The NASIS Data Model Diagrams use Bachman notation, one of several competing standards for symbols used in a data model. The NASIS Data Model Diagram document contains the following:

One Conceptual Entity Relationship Diagram for NASIS as a Whole

One “High Level” Physical Data Model for NASIS as a Whole

A Physical Data Model for Each NASIS Owned Object (see Owned Objects)

One Summary “Load Related/Find Related” Physical Data Model for NASIS as a Whole

One Primer on Bachman Notation

Naming Conventions

Logical Names, Physical Names and Labels

In NASIS, some database entities have more than one name. Tables and columns/attributes have a logical name, a physical name and a label. The logical name is a longer name, usually around 30 characters, and is usually a less cryptic business oriented name. Physical names are the names used in the actual physical NASIS database. Physical names are the names a person would use to make a raw SQL query (a non-NASIS Query) against a NASIS database. In Informix On-Line, the database management system in which NASIS is implemented, physical names are restricted to a maximum of 18 characters. Either logical or physical names may be used in developing NASIS Queries, Reports and Properties. Labels are what are displayed above columns and at the top of a table’s spreadsheet window in NASIS. Logical names, physical names and labels are documented in both the NASIS Template Model reports (see NASIS Template Model Reports) and in the NASIS on-line help system.

Domains (Choice Lists) have only a logical name. Indexes have only a physical name. In the NASIS Template Model reports, database entity names are usually labeled unambiguously. If a name label on a report does not explicitly indicate whether it is a logical name, physical name or a label, assume the name is a logical name, except in the case of an index, which only has a physical name.

Modal Attributes

Modal attributes are used to express the range in variability of an attribute of the soil throughout a component of a map unit. This variability is expressed as a range (a low value and a high value) with an accompanying representative (expected) value. Therefore, a modal attribute requires three actual physical columns. In the NASIS Template Model reports, modal attributes are listed as a single column, even though in a physical NASIS database, a modal attribute is implemented as three physical columns. In the Table Structure Report, the column titled “Exp. Var.” (expression of variability) indicates whether or not a column is modal.

Each column in a database table must have a unique name. The physical name of a modal column in a NASIS Template Model report resolves to three unique names in a physical NASIS database by adding an extension that makes each name unique. The extension “_h” is used for

the high value. The extension "_l" is used for the low value. The extension "_r" is used for the representative value. For example, the attribute with the logical name "bulk_density_fifteen_bar" is a modal attribute whose physical name in a NASIS Template Model report is "dbfifteenbar". In the table where this attribute occurs, there are three columns, "dbfifteenbar_h", "dbfifteenbar_l" and "dbfifteenbar_r".

Calculable Attributes

Since NASIS 3.0, an attribute may have an associated routine that can be used to automatically calculate its value based on the value of one or more other attributes. Such an attribute has an associated column that indicates if the current value of that column was calculated (status code = "C"), manually entered since being able to be calculated (status code = "M"), manually entered prior to being able to be calculated (status code = "P") or if the origin of that value is unknown (status code is Null). In the NASIS Template Model reports, a calculable column is listed as a single column, even though in a physical NASIS database, a calculable column is implemented as two physical columns. In the Table Structure Report, the column titled "Calc?" indicates whether or not a column is calculable.

The rationale for status "P" (manually entered prior to being able to be calculated) may not be obvious. Over time, attributes that could only be manually entered have become calculable. When an attribute becomes calculable, its initial status code is set to "P", when its corresponding value is not null. The reason for doing this is so that a user can then request calculations for a number of instances of that attribute, where calculated values won't override existing values without explicit authorization from the user. By providing status "P" in addition to status "M", a user can choose when to override which manually entered values with a calculated value.

The name of a calculable attribute's associated calculation status column is the physical name of that column, with a "_s" (s = source) appended to the end of the name. For example, the attribute with the logical name "taxonomic_classification_name" is a calculable attribute whose physical name in a NASIS Template Model report is "taxclname". In the table where this attribute occurs, there are two columns, "taxclname" and "taxclname_s".

It is possible for a modal attribute to be calculated. In this case both modal and calculation status extensions are appended to the end of the physical name of such a column. For example, the attribute with the logical name "aashto_group_index" is a calculable, modal attribute whose physical name in a NASIS Template Model report is "aashind". In the table where this attribute occurs, there are six columns, "aashind_l", "aashind_ls", "aashind_r", "aashind_rs", "aashind_h" and "aashind_hs".

NASIS Site ID Attributes

NASIS Site ID's are assigned by the NASIS Hotline when a new NASIS site is created. A NASIS Site ID, which is an integer number, uniquely identifies a particular NASIS site. Prior to NASIS 5.0, a NASIS site represented both an organizational unit (typically an MLRA office) and a physical database, since each NASIS site had its own physical NASIS database. Since NASIS 5.0, NASIS is a centralized database. A NASIS site now represents only an organizational unit, not a physical database, since data for all NASIS sites now resides in the same physical database.

There is a NASIS site for every MLRA office. There are also a number of special NASIS sites that were created for special purposes. Two examples are the NASIS sites "pangaea" and "flora". The NASIS site "pangaea" is responsible for developing nationally used Queries, Reports, Properties, Evaluations, Rules, Calculations and Edit Setups. The NASIS site

"pangaea" is also responsible for the management of the common set of geographic areas and geomorphic feature terms. The NASIS site "flora" is responsible for the management of the common set of NRCS ecological sites and the official set of plants recognized by NRCS.

NASIS Site ID attributes end in either the substring "database_iid_ref" (logical name), "dbiidref" (physical name). NASIS Site ID column names are always based on a particular owned object. For example, the NASIS Site ID column name for the root table of the Legend owned object is "legend_database_iid_ref". A NASIS Site ID name always include the substring "ref" because every NASIS Site ID is really a foreign key reference to the NASIS Site table.

Since NASIS 5.0, NASIS Site ID attributes occur only in the root table of an owned object (see Owned Objects). This means that there are only 25 NASIS Site ID attributes in NASIS, since there are 25 owned objects in NASIS. Even though a NASIS site now represents only an organizational unit, a NASIS site is still the entity by which access to data is controlled. This is still appropriate because MLRA offices still have primary responsibility for the data assigned to them.

A NASIS Site ID value is assigned when a new instance of an owned object is created. The value of a NASIS Site ID column can only be changed in one of two ways. When an instance of an owned object is pasted, its ownership is always assigned to the NASIS site corresponding to the NASIS user's current default group. The only other way that the value of a NASIS Site ID column can be changed is via the "Change Ownership" function of the NASIS software. The value of a NASIS Site ID column cannot be changed via the direct edit capabilities of the NASIS software.

Record ID Attributes

Every table in NASIS has one Record ID column that is used to uniquely identify records in that table. Such a column is referred to as the table's "identifying" Record ID column. Since NASIS 5.0, this column alone is the primary key of its corresponding table. Prior to NASIS 5.0, when NASIS was still a distributed database, each table's primary key consisted of two columns, a NASIS Site ID and a Record ID. This was necessary because it was not feasible to generate globally unique (unique among ALL NASIS sites) Record ID's.

While each table in NASIS has one identifying Record ID column that serves as that table's primary key, a table may contain columns corresponding to Record ID's from other NASIS tables. These columns are referred to as "foreign keys", i.e. they uniquely identify a record in a related table. This is how relationships are established in a relational database (see Referential Integrity).

The name of Record ID attributes end in any of the substrings shown below. Record ID attribute names, unlike NASIS Site ID attribute names, do not contain the substring "database" (logical name) or "db" (physical name).

_iid (logical name)

iid (physical name)

_iid_ref (logical name)

iidref (physical name)

Record ID's that end in "_iid" or "iid" are attributes that serve as the primary key of a table. Record ID's that end in "_iid_ref" or "iidref" are attributes that serve as a foreign key of a related

table. A table may have only one primary key but may have multiple foreign keys (see Record Identification and Referential Integrity).

Record ID column names are always based on a particular table. For example, the identifying Record ID column name for the Legend Staff table is “legend_staff_iid”. In making a foreign key reference to the Legend Staff table, the column name “legend_staff_iid_ref” is used as the foreign key.

A Record ID value is assigned when a new instance of a record is created. The value of an identifying Record ID column cannot be changed, per se. When an instance of a record is pasted, it always represents a new record, and therefore a new identifying Record ID value is assigned. The value of an identifying Record ID column cannot be changed via the direct edit capabilities of the NASIS software.

Index Naming Conventions

In NASIS an index name is usually a contraction of its table’s physical name with an extension attached. The extension “_p” implies that the index is the primary key of the table. An extension that contains the substring “_c” implies that the index corresponds to a foreign key of a related table. The letter “c” corresponds to “child”, as in “child table”. Any characters following the “_c” are used to distinguish one foreign key from another, since a table may have more than one foreign key. The extra characters in the foreign key extension imply the name of the related table. For example the extension “_cd” usually means “child of database” (the table that records a NASIS site used to be referred to as the “database” table). An extension that contains the substring “_uv” implies an alternate unique key or candidate key. The letters “uv” correspond to “user view”. User view keys are combinations of columns that must be unique from a soil survey business perspective (see Record Identification). The extension “_d” implies that the index is a link display index (see Links). These conventions are generally applicable, but not absolute.

Exceptions to the Naming Conventions

While the entire NASIS database uses roughly the same naming conventions, the conventions are less rigorously applied to the non-editable data dictionary tables.

Data Types

Data Type Summary

For NASIS we define three different data types, logical, “physical” and the ultimate Informix On-Line data type. Logical data type is a conceptual data type that is assigned independently of the software in which a system will ultimately be implemented. Physical data type takes into account the software and database management system in which a system will ultimately be implemented. For NASIS, physical data types include not only the native data types supported by Informix On-Line but also some custom data types that have meaning only to the NASIS software. In NASIS, all physical data types are ultimately implemented in some native Informix On-Line data type. The following table shows the relationship between the different data types defined for NASIS.

Logical Data Type	Physical Data Type	Ultimate Informix On-Line Data Type
Boolean	Boolean	Smallint
Choice	Ordered Code	Smallint
Choice	Unordered Code	Smallint
Date/Time	Date	Date
Date/Time	Datetime	Datetime
Edit Setup	Edit Setup	The actual data type of this column in a physical NASIS database is “text”, but its value is null. It is merely used as a location from which the Edit Setup Editor may be invoked. An Edit Setup is actually implemented as a number of hidden tables with columns of various data types.
Evaluation	Evaluation	Text
Float	Decimal	Decimal
Float	Float	Float
Float	Smallfloat	Smallfloat
Integer	Decimal	Decimal
Integer	Integer	Integer
Integer	Serial	Serial
Integer	Smallint	Smallint
Money	Money	Money
Property	Property	Text
Query	Query	Text
Rule	Rule	Text
String	Character	Character
String	Varchar	Varchar
Vtext	Text	Text

Data Type Specifics

Some of the less obvious data types are defined below.

Boolean

In NASIS, Boolean values are displayed as “Yes” and “No”, and are recorded in the actual physical NASIS database as 1 and 0 respectively.

Choice

Attributes whose data type is “choice” have their domains restricted to a finite list of choices, where each choice in the domain is represented by a unique ASCII string that may contain up to 128 characters. Each choice in such a domain is also assigned a unique integer number and a unique label. In the actual physical NASIS database, the integer number associated with a choice, rather that the ASCII string or label associated with a choice, is what is stored in the database (see Class Attributes).

Edit Setup

The Edit Setup "data type" was introduced in NASIS 4.0. Like a number of other complex NASIS data types, we established this data type so that when a NASIS user puts the cursor in the cell of an attribute with this data type, we know to enable the zoom button, as well as what special editor to invoke when a NASIS user hits the zoom button.

The column in the Edit Setup table whose data type is "Edit Setup" actually contains no data whatsoever. The ultimate Informix On-Line data type of such a column in a physical NASIS database is "text", but its value is null. This column is merely used as a location from which the Edit Setup Editor may be invoked. An Edit Setup owned object is actually implemented using a number of tables that are not viewable or directly editable via the normal NASIS spreadsheet paradigm.

An Edit Setup is a set of specifications for the following.

> Which columns in a table should be displayed in that table's corresponding spreadsheet window?
>
> In what order should the selected columns of a table be displayed?
>
> What is the display width of each of the selected columns?
>
> How many of the left most selected columns of a table should be frozen (not subject to scrolling)?
>
> On what column or columns should the records in a table be sorted?

Prior to NASIS 4.0, a NASIS user had no control over any of these choices.

Evaluation

An Evaluation is the function that transforms the value of a Property (something whose data type is "property") into a value somewhere within the zero to one range of the fuzzy logic domain. An Evaluation is part of the NASIS fuzzy logic interpretation generation system. For a discussion of interpretations in NASIS, see the NASIS on-line help system. This data type is implemented as a text field that records the actual logic used to make this transformation. A special Evaluation editor is opened in order to create or modify an Evaluation.

Property

A Property is a virtual attribute whose value is calculated on demand when either a report or an interpretation is generated. For a discussion of interpretations in NASIS, see the NASIS on-line help system. Property values are not permanently stored in a NASIS database. This data type is implemented as a text field that actually contains the NASIS Property source language recognized by the NASIS Property generator. The source language is simply ASCII text that may be viewed and edited in NASIS.

Query

A Query is used to select data that meets certain criteria from a NASIS database, for the purpose of manipulating or viewing that data. This data type is implemented as a text field that actually contains the NASIS query source language recognized by the NASIS query processor. The NASIS SQL syntax is very similar to, but not exactly the same as, standard SQL syntax. At runtime, a NASIS Query is ultimately translated to true SQL. A special Query editor is opened in order to create or modify a Query.

Rule

A Rule is a function that transforms an expression of Evaluations and/or Rules into a value somewhere within the zero to one range of the fuzzy logic domain. A Rule is part of the NASIS fuzzy logic interpretation generation system. For a discussion of interpretations in NASIS, see the NASIS on-line help system. This data type is implemented as a text field that records the actual logic used to make this transformation. A special Rule editor is opened in order to create or modify a Rule.

Vtext

This data type is used to record variable length narrative text that may contain paragraph breaks.

Class Attributes

Class Attribute Overview

A class attribute is one that has a domain or choice list associated with it. The value that a class attribute may assume is restricted to a finite set of values. In NASIS, an integer number is associated with each choice in a domain. When a choice is recorded in a NASIS database, the integer number associated with that choice, rather than the string representing the choice itself, is stored in the database. This scheme reduces the space required for a NASIS database, and makes it possible to change the spelling of codes without having to change the underlying NASIS database. This scheme also protects against having to change the field width of a class attribute column when a new choice that contains more characters than the current longest choice is added to a domain, subject of course to the restriction than no choice may have more than 128 characters.

A Choice List ID uniquely identifies each domain or choice list. Each choice in a domain has five attributes associated with it, Choice Sequence, Choice ID, Choice, Choice Label and Choice Description. For each domain, Choice Sequence, Choice ID, Choice and Choice Label are required fields. Within a particular domain, each Choice ID, Choice and Choice Label must be unique. Choice labels were not required, or required to be unique, prior to NASIS 4.0.

Choice sequence orders the choices in a domain for display in a choice list. Domains are either explicitly (manually) ordered or are sorted by Choice. This concept is referred to as “domain ordering”. There is also a concept of an “ordered” domain. An “ordered” domain is one whose choices represent a ranking order from lowest to highest. For “ordered” domains, Choice Sequence is overloaded to specify both the display sequence and a choice’s rank. It is not

possible to display an “ordered” (ranked) domain from highest to lowest, since both the display order and rank order are specified by the same attribute. An “ordered” domain has a physical data type of “Ordered Code”. An unordered domain has a physical data type of “Unordered Code”.

Each Choice ID, Choice and Choice Label must be unique within a given domain. Choice ID is an integer number. Choice is typically a lower case ASCII string whose length is always restricted to a maximum of 128 characters. To manually enter a choice in NASIS, as opposed to selecting a choice from a choice list, one must enter the value associated with Choice. Choice Label is typically a mixed case ASCII string whose length is always restricted to a maximum of 252 characters. Choice Labels are typically output on reports. Choice Description is a narrative text description of a particular choice in a domain. Below is an example NASIS domain.

Domain: Observation Method				
Choice Sequence	**Choice ID**	**Choice**	**Choice Label**	**Choice Description**
1	3	auger, bucket	Bucket Auger	Sample extracted by means of open, sand, closed, of mud bucket auger. Generally 5 to 12 cm diameter.
2	4	auger, screw	Screw Auger	Sample extracted by means of external thread hand auger, or mechanically powered flight auger. Generally 2 to 30 cm diameter.
3	7	cut	Cut	Sample extracted from a relatively large near vertical cut such as a roadcut. Generally greater than 4 m in length.
5	8	pit, large or quarry	Large Pit or Quarry	Sample extracted from a large open pit or large very vertical bank, such as borrow pit, quarry, or stream cutbank. Generally greater than 33 m in length.
6	5	pit, small	Small Pit	Sample extracted from a small hand-dug pit, dug with a shovel and/or hand pick. Generally less than 1m x 2m in size.
7	1	push tube	Push Tube	Sample extracted with a push tude, either hand held or hydraulically powered. Generally about 2 to 10 cm in diameter.
8	2	shovel slice	Shovel Slice	Shovel extracted by means of an undisturbed slice of soil with a shovel (sharpshooter, spade) from the side of a small pit. Generally about 20 x 40cm is size.
9	6	trench	Trench	Sample extracted from the wall of a trench or pit dug with the aide of a backhoe. Generally larger than 1 x 2m in size.

In NASIS, when you open a choice list for a class attribute, Choice is displayed on the left, and if a second column is visible to its right, that column contains Choice Label. When you select a choice from a choice list, Choice is displayed in the cell of the NASIS spreadsheet window. When a choice is written to the database, Choice ID is the value that is actually stored to the database. The NASIS software translates Choice ID's to Choices when data is read from the database and displayed in a table's spreadsheet window, and the NASIS software translates Choices to Choice ID's when a NASIS user posts edit changes to the database.

In NASIS, class attributes have a logical data type of "Choice" and a physical data type of either "Ordered Code" or "Unordered Code". The name of the domain associated with such an attribute can be obtained from the Attribute Report. A listing of the choices that make up a domain are available in the Choice List Report. The Choice List Report shows the Choice Sequence, Choice ID, Choice, Choice Label and Choice Description associated with each choice in a domain as well as the ID associated with the domain as a whole (Choice List ID).

Class Attributes and Raw SQL Queries (non-NASIS Queries)

A query against a class attribute requires either a direct or indirect reference to the Choice ID associated with the particular choice on which you wish to select. The ultimate retrieval of either the Choice, Choice Label or Choice Description associated with a class attribute requires either a direct or indirect reference to the Choice List ID of the domain to which a class attribute's values are restricted. Writing queries involving class attributes requires references to tables and columns in the operational data dictionary.

Querying Class Attributes Using Direct References

The Choice List Report is required in order to query class attributes using direct references. Both Choice ID and Choice List ID can be found in the Choice List Report. The Attribute Report must be used to determine the choice list associated with a class attribute. Using direct references is the easiest way to query class attributes.

This example shows a query against a class attribute where neither the Choice or Choice Label values associated with any class attributes are actually retrieved. This query requires prior knowledge of the Choice ID on which you wish to select. In NASIS, the value displayed for a class attribute is the value referred to as Choice. The Choice ID associated with a particular Choice can be determined from the Choice List Report.

In the following examples, that part of the SQL statement that is not variable is in lower case. The part of the SQL statement where something appropriate must be substituted, is in upper case and italicized.

select *WHATEVER*
from *TABLE_PHYSICAL_NAME*
where *TABLE_PHYSICAL_NAME.CLASS_ATTRIBUTE_PHYSICAL_NAME = CHOICE_ID*

The next example shows a query where the Choice and Choice Label values associated with a class attribute are retrieved. This query requires prior knowledge of the Choice List ID of the choice list associated with the class attribute whose Choice and Choice Label values you wish to retrieve.

select *WHATEVER*, dd_codes.code_nm, dd_codes.code_label
from *TABLE_PHYSICAL_NAME*, dd_codes
where *TABLE_PHYSICAL_NAME.CLASS_ATTRIBUTE_PHYSICAL_NAME = CHOICE_ID*

and *TABLE_PHYSICAL_NAME.CLASS_ATTRIBUTE_PHYSICAL_NAME* = dd_codes.code_val
and dd_codes.code_dom_iid = *CHOICE_LIST_ID*

Querying Class Attributes Using Indirect References

Class attributes may be queried using indirect references to Choice ID's and Choice List ID's. These types of queries can be made without the aid of the Choice List Report, so long as one has prior knowledge of the value of each Choice (the shorter ASCII string as opposed to an integer number) on which they wish to select. Using indirect references is not the easiest way to query class attributes.

This example shows a query against a class attribute where neither the Choice or Choice Label values associated with any class attributes are actually retrieved. This query requires prior knowledge of the Choice on which you wish to select. In NASIS, the value displayed for a class attribute is the value referred to as Choice.

In the following examples, that part of the SQL statement that is not variable is in lower case. The part of the SQL statement where something appropriate must be substituted, is in upper case and italicized.

select *WHATEVER*
from *TABLE_PHYSICAL_NAME*
where *TABLE_PHYSICAL_NAME.CLASS_ATTRIBUTE_PHYSICAL_NAME* =
(select code_val
from dd_codes
where code_nm = "*CHOICE*" and
code_dom_iid =
(select unique elm_dom_iid
from dd_elm
where elm_imp_nm = "*CLASS_ATTRIBUTE_PHYSICAL_NAME*"))

The next example shows a query where the Choice and Choice Label values associated with a class attribute are retrieved. This query requires no prior knowledge of any Choice ID or Choice List ID.

select *WHATEVER*, dd_codes.code_nm, dd_codes.code_label
from *TABLE_PHYSICAL_NAME*, dd_codes
where *TABLE_PHYSICAL_NAME.CLASS_ATTRIBUTE_PHYSICAL_NAME* =
(select code_val
from dd_codes
where code_nm = "*CHOICE*" and
code_dom_iid =
(select unique elm_dom_iid
from dd_elm
where elm_imp_nm = "*CLASS_ATTRIBUTE_PHYSICAL_NAME*"))
and *TABLE_PHYSICAL_NAME.CLASS_ATTRIBUTE_PHYSICAL_NAME* = dd_codes.code_val
and dd_codes.code_dom_iid =
(select unique elm_dom_iid
from dd_elm
where elm_imp_nm = "*CLASS_ATTRIBUTE_PHYSICAL_NAME*")

Exceptions to the Class Attribute Conventions

The only exceptions to the general class attribute conventions are certain data dictionary class attributes for which the choice itself (Choice), rather than Choice ID, is stored in the database. In these cases, the choices in the domain may not even be assigned Choice ID's.

Indexes

NASIS defines a multitude of indexes, both real (physical) and imaginary. In addition to their standard uses, NASIS uses indexes as a convenient shorthand way of specifying an ordered subset of columns from a table. Real indexes are used to enforce uniqueness constraints (see Record Identification), improve database performance and to establish referential integrity constraints (see Referential Integrity). Imaginary indexes are used to record the column or columns that are displayed in place of a link column (see Links).

Record Identification

Record Identification Overview

One tenet of relational database design is that every table should have a key. A key is a column or set of columns whose value or combination of values must be unique within that table. Informix On-Line does not enforce this tenet on a user. Each table in an Informix database does have an Informix generated "Row ID" column that Informix uses to uniquely identify each record in the table. This column is managed exclusively by Informix, and a record's Row ID can change when an Informix database is compressed. The NASIS design standards require that each table have a key other than "Row ID". NASIS uses two record identification schemes, a system oriented scheme and a business oriented scheme.

In NASIS, records are primarily identified through the use of surrogate (system generated) keys, as opposed to the more traditional scheme of natural keys. A natural key is one or more user editable columns in a table whose value or combination of values must be unique within that table, where the columns involved in that constraint pertain to some business rule of the system being modeled. Few the tables in NASIS have any natural key defined. Unfortunately, for the most part, a NASIS user is on their own to make sure that a table does not contain duplicate records (apparently duplicate from the user's point of view, since each record in a NASIS database is always unique from the system's point of view).

Although we use surrogate keys in NASIS, there are a variety of problems with surrogate keys. One problem with surrogate keys is that it is possible to have two "different" records (at least from the software's point of view) that logically represent the same instance of some entity. Another problem with surrogate keys is that sometimes, when data is reorganized or converted between NASIS releases, a surrogate key value may change, yet the entity identified by that new key still represents the same logical construct that the old key identified. Yet another problem with surrogate keys comes up when a record is accidentally deleted and must be manually re-added to the database. In order to restore the "same" instance, the NASIS user must know what the previous surrogate key value was, and then must be able to assign that previous surrogate key value to the re-added record. In NASIS, a user cannot specify the surrogate key value for a newly inserted record. In NASIS, the assignment of surrogate key values is handled exclusively by the NASIS software.

There are a number of reasons that NASIS defines almost no true natural keys. Prior to NASIS 5.0, NASIS was a distributed database (data segmented between multiple physical locations). For a distributed database, it is difficult to manage natural key conflicts among all of the databases of a distributed system in real time, especially when not all of those databases may not even be up and operating at the same time.

Even though NASIS is now a centralized database, making natural keys more feasible, we have other problems in defining natural keys for our tables. The first problem that we have is that many of our candidate natural keys have one or more potentially null columns. One example of this is the attribute correlation date, which is a candidate for inclusion in the natural key that uniquely identifies a particular legend. Unfortunately, the record for a legend is established long before the correlation date is known. In general, for a variety of reasons that are beyond the scope of this document, no key column should ever be able to be null.

The second problem we have in defining natural keys for our tables is that many of our candidate natural key columns are attributes that, by design, are expected to be periodically updated over the lifetime of the entity they identify. These include such attributes as legend status and map unit status. A good candidate natural key is made up of columns whose values are not expected to change over the lifetime of the entity that the key identifies.

A third problem we have in defining natural keys for our tables is that in some cases, the attributes that distinguish one instance of an entity from another instance of that entity do not all reside in the root table of that entity. The candidate columns for a table's key are restricted to the columns that reside in that table. In other words, one cannot define a key for a table that includes columns from some other table. An example where we have this problem is in the case of a map unit component. Some of the attributes that may legitimately be used to distinguish one component from another reside in tables that are children of the component table (soil texture for example).

System Oriented Record Identification Scheme

Switching from a distributed database to a centralized database made a huge impact on our internal key scheme. Prior to NASIS 5.0, the system identified a record by the combination of a NASIS Site ID and a table Record ID. This was necessary because the software could only ensure the uniqueness of a Record ID within the context of a particular NASIS site, which at that time corresponded to one of a number of physical databases (see NASIS Site ID Attributes and Record ID Attributes).

Now that NASIS is centralized, there is only one physical database in one physical location. Because there is only one physical database, we can ensure globally unique Record ID's. Therefore, the primary key of almost every table in NASIS is that table's identifying Record ID column.

Each owned object has a single root table, and since the root table's Record ID is sufficient to uniquely identify every record in that table, an owned object's root table Record ID value is therefore sufficient to uniquely identify an instance of that owned object.

The phrase "a table's identifying Record ID column" was used earlier in this discussion. A table's "identifying" Record ID column is a special column whose value is automatically supplied when a new record is created. The NASIS software supplies this value. A NASIS user cannot specify this value when a record is created, nor can a NASIS user ever directly edit the value of a Record ID column. The only time the value of a Record ID column is updated is when a record is copied and then pasted. Every pasted record is, by definition, a new record, a NASIS always supplies a new Record ID value for a newly pasted record.

Since the values of identifying Record ID columns are managed exclusively by the NASIS software, there is no possibility of a key clash of NASIS surrogate keys, at least if the software is functioning properly.

Keep in mind that a table may contain non-identifying Record ID columns in addition to the table's identifying Record ID column. Such columns are referred to as "foreign keys". The value of such a column uniquely identifies a record in a related table. This is how relationships are established in a relational database, which NASIS happens to be (see Referential Integrity).

Business Oriented Record Identification Scheme

Another reason that a column or combination of columns is constrained to be unique is to enforce a business rule. Since each table in NASIS already has a system oriented primary key, any additional business oriented key constitutes an alternate unique key or candidate key. In the NASIS documentation, we sometimes refer to business oriented keys as "user view" keys. A better term is business rule keys. However, the name of an index that establishes a business rule key typically contains the substring "uv" (user view) rather than "br" (business rule).

Many of the business rule keys in NASIS are a combination of business oriented columns and internal system, or surrogate key, columns. There are only a handful of true natural keys in NASIS, and none of those natural keys pertain to any of our major soil entities (legend, map unit, component, horizon, site, pedon, transect, etc.).

NASIS manages its system, or surrogate, keys so that surrogate keys clashes can never occur. Since business rule keys are managed exclusively by NASIS users, business rule key clashes CAN occur. One place where business rule key clashes can occur is during data entry and data editing. It is possible to enter values for a new record that will cause a key clash with an existing record. It is also possible to edit an existing record such that its updated values will cause a key clash with an existing record.

Another less obvious place where business rule key clashes can occur is when records are copied and pasted. If a record that contains a business rule key is copied and pasted, by definition, one of those two records is going to have to be edited before data can be committed, since the very act of pasting the copied record creates a key clash.

Referential Integrity

Referential Integrity Overview

In NASIS, tables are related exclusively through the use of surrogate keys, as opposed to the more traditional scheme of relating tables through the use of business oriented or natural keys. The relationship between a parent table and a child table is established by migrating the parent table's primary key to the child table. In NASIS, every table's primary key is independent of its parent table's primary key. This means that the primary key of a map unit record does not include the columns that identify that map unit's corresponding soil survey area. One ramification of this is that the number of columns in a primary key does not increase with table depth in a hierarchy. The vast majority of the relationships in NASIS represent joins on a single column, since the primary key of most tables in the NASIS database employ a single column primary key.

This scheme, while simple and efficient, causes some confusion to our users who were familiar with the State Soil Survey Database (SSSD), the information system that NASIS replaced. In the Layer table in SSSD you could look at a record and know its corresponding soil survey area, map unit and component. (Well, sort of. When it comes to component, SSSD, like NASIS, used a relatively meaningless number to identify a component.) This was because SSSD not only tended to use natural keys to identify its records, it migrated those keys all the way down a hierarchy. In SSSD, you could select all Layer records for a given soil survey area without having to join the Layer table to any other table.

In NASIS, the Component Horizon table corresponds to what was the Layer table in SSSD. In the Component Horizon table there are no columns that identify that horizon's corresponding map unit or soil survey area. This is because is NASIS, keys are not migrated down a hierarchy. In the Component Horizon table there are columns that identify its corresponding component, but those columns are relatively meaningless integer numbers. This is because in NASIS, entities are identified by surrogate keys as opposed to being identified by more meaningful natural keys (see Record Identification). In NASIS, selecting all horizon records for a given soil survey area requires a join that includes all tables in the hierarchy from Legend to Component Horizon.

Because none of the columns that participate in the relationship between two tables are editable, a NASIS user cannot make an edit to an existing record that ever requires an update of that record's primary key. The only operation a NASIS user can perform that requires updating of a record's primary key is a paste operation. Since a paste operation creates a new instance of a record, that record cannot have the same primary key as the original source record.

Informix On-Line provides the capability to enforce referential integrity at the database engine level. Enforcing referential integrity at the database engine level means that Informix itself enforces the rules, independent of the NASIS software. Since NASIS 3.0, referential integrity has been enforced at the database engine level for all relationships in a NASIS database.

When referential integrity is enforced, whether at the database engine level or via the NASIS software, the following rules are enforced.

1. When a record is added to a dependent table and values are specified for the foreign key of a related table, a record with that key must already exist in the related table.

2. You cannot delete a record from a table if any related records exist in one of its dependent tables, unless the delete rule for the relationship between those two tables is "cascade" (as opposed to "fail" or "restrict").

The Relationship Report shows what tables are related, and the names of the columns on which two tables are joined. In this report, the related tables are shown a pair at a time. The table listed to the left is the table on the "left" side of the one to many or one to one relationship between the two tables. The table listed to the right is therefore on the "right" side of the relationship. (The vast majority of relationships in NASIS are one to many. There are only a couple of one to one relationships in the NASIS database.) The columns on which the two related tables are joined are listed as two separate indexes. Column N in one index joins to column N in the other index.

Relationship Categories

Relationships in NASIS are divided into two broad categories, edit relationships and link relationships. An edit relationship is a relationship between two tables in the same owned object that may be traversed via the "Down Table" and "Up Table" functions of the NASIS software. A link relationship is a relationship between two tables, that may or may not be part of the same

owned object, that cannot be traversed via the “Down Table” and “Up Table” functions of the NASIS software. Most link relationships may be used and traversed via the “Load Related” and “Find Related” functions of the NASIS software. Not all link relationships in a NASIS database are traversable. When the same two tables share more than one link relationship, only one of those link relationships is currently traversable. At some point in the future, NASIS may be enhanced such that all relationships between two particular tables may be traversed.

Non-Typical Relationships in NASIS

In most cases, the relationship between a parent table and a child table is based on migrating the parent table’s primary key to the child table. When two tables in NASIS are related on more than one column, the extra columns are typically necessary in order to constrain a lookup list of parent table records to a subset of the complete set of parent table records. This occurs when a higher level relationship constrains the set of possible relationships at a lower level in the data model hierarchy. For example, when displaying a list of map units in the Mapunit Area Overlap table, that list must be constrained to the set of map units corresponding to the legend that was selected in the parent Legend Area Overlap table.

Referential Integrity and Raw SQL Queries (non-NASIS Queries)

A query that requires a join between the Data Mapunit, Component and Horizon tables would look something like the following. The information required to write such a query can be obtained from the Table Structure Report (table and column physical names) and the Relationship Report (the logical names of the join columns). It is necessary to translate the logical names of the tables and join columns obtained from the Relationship Report to their corresponding physical names by examining the Table Structure Report.

select *WHATEVER*
from datamapunit, component, chorizon
where component.dmuiidref = datamapunit.dmuiid
and chorizon.coiidref = component.coiid

Links

We have included a discussion about links because it helps in understanding why a table’s corresponding spreadsheet window may not display exactly the same columns that exist in the underlying table. A link is a column or columns in a table used to record the primary key of a record in a related table. A link is the same thing as a foreign key. In NASIS, all tables are related through the use of surrogate or system generated keys, which are always integer numbers. Showing these internal numbers in a table’s corresponding spreadsheet window would not be very informative. Instead of displaying a foreign key, soil business related attributes from the related table are displayed in lieu of the meaningless integer foreign key columns.

More than one column may be displayed in place of the original link column. Before NASIS 3.1, when more than one column was displayed in place of a link column, only one of those columns could actually be used for data entry in a table’s corresponding spreadsheet window. Since NASIS 3.1, a NASIS user may open a selection list for any column of a multi-column link display index.

For example, the link relationship between Local Plant and Plant displays 3 columns from the Plant table; Plant Symbol, National Vernacular Name and Scientific Name. When the selection

list is opened from the Plant Symbol column, the list will be sorted by Plant Symbol. When the selection list is opened from the National Vernacular Name column, the list will be sorted by National Vernacular Name. The 3 columns in the selection list will always appear in the same left to right order, but the sorting of the list depends on the column from which the selection list was opened. The selection list search capability is also based on the column from which the selection list was opened.

The choice list that is displayed for a link column looks very similar to a choice list that is displayed for selecting a domain choice for a class attribute (see Class Attributes). The difference is that the data in the choice list for a link column comes from some NASIS table other than the table (dd_codes) where ALL static domains are recorded. In most cases, the table that provides the choices for a link column lookup can be viewed and edited in NASIS, at least by some users. The table that provides the choices for all class attributes (dd_codes) cannot be viewed or edited in a NASIS spreadsheet window.

Being a link column is considered to be an “attribute” of a column, as far as the NASIS data dictionary and software is concerned. Link columns (columns that make up all or part of a foreign key) are listed in the Link Report. The Link Report contains the information that specifies how a link is resolved. Resolving a link is the process whereby the soil business related attributes from a related table are displayed in place of the original link column. The table containing the link column (foreign key) is shown to the left. The label of the link column in this report is “Column Logical Name”. The table containing the link lookup data is shown to the right. Underneath these two sections, the contents of three indexes are displayed. The first two side by side indexes show the columns on which the two related tables are joined (see Referential Integrity). The last index (under the right join index) shows the column or columns that will be displayed in place of the original link column. We refer to this index, which is usually not a physical index, as the “Lookup Table Display Index” or the “Link Display Index”.